

Application No. 09/539,197

IV. OBJECTION TO SPECIFICATION

Applicants submit a substitute specification pursuant to 37 CFR § 1.125 and MPEP § 608.01 (q) that replaces original pages 1-45, which all include the header "003048.P011". The substitute specification is accompanied by a marked up copy showing the changes between the original application, as filed, and the substitute specification, to illustrate deletion of the header from each page, as requested by the Examiner. Other minor typographical errors have also been corrected. The substitute specification does not contain any new matter and includes the same changes as are indicated in the marked up copy. Applicants respectfully request that the substitute specification be entered in this case and that the objection to the specification be withdrawn.

V. REJECTION UNDER 35 U.S.C. §102(b)

A. The Office Action Rejection

In pages 4-5 of the Office Action, claims 1-44 were rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 6,205,560 to Hervin, et al. ("Hervin").

The Applicants respectfully traverse the rejection and submit the following remarks to expedite prosecution of the application.

B. Claims 1-44 are Patentable Over Hervin

Applicants respectfully submit that claims 1-44 are not anticipated by Hervin since the cited reference does not disclose or suggest various aspects of Applicants' claims.

The subject application generally discloses a method and system for debugging an executing service by setting a breakpoint within an executing service and saving a minimum state of the executing service. A program counter of the executing service is altered. The program counter and state of the executing service are restored. More specifically, a number of executing services 610 may run within a basic functional unit or PE 670. The PE 670 includes a save stub 662 and a restore stub 664. The save stub 662 is an executable program written to save the minimum state of construct 610. The restore stub 664 restores the minimum state from memory 620. p. 18, lines 8-12. The minimum PE state (MPES) is a minimum amount of executing service 610 state registers which are saved and restored in order to halt service 610 execution and restart it again without altering the functional behavior of the service. p. 18, lines 10-14.

The Hervin patent is discloses a circuit for diagnosing and debugging a processor. A set of debug registers 42 (as compared to registers in a register file 24) identify break points in data or instruction addresses during diagnosing and debugging of a processor. col. 4, lines 33-35. More specifically, four debug

Application No. 09/539,197

registers contain linear memory or instruction addresses for one of four possible breakpoints. col. 6, lines 40-57. The size of the linear addresses can be 1, 2, or 4 bytes and can be an address of data in memory or to an instruction address of a program. col. 6, lines 42-44. Different numbers of registers can be used for setting breakpoints. col. 6, lines 55-57. Memory addresses in the debug registers are "data" breakpoints, whereas instruction addresses in the debug registers are "instruction" breakpoints. col. 6, lines 45-51.

The Hervin patent further discloses that an instruction is fetched and decoded by an instruction decoder. The processor compares the linear instruction address to the breakpoint addresses in the debug registers. If the addresses match, then a debug fault exception is generated and a software debug interrupt handler controls the processor. If the addresses do not match, then data addresses for the instruction are calculated, and the data addresses are compared to breakpoint addresses in the debug registers. The instruction is then executed. If a match of the data addresses occurred, a debug trap exception is generated, and control of the processor is switched to the software debug interrupt handler. If there is no data address match, then no exception is generated. col. 8, lines 29-43. A breakpoint, either a data or instruction address, is written to one of the debug address registers. A breakpoint counter counts down from a predefined number X, so that the exception does not occur until the Xth occurrence of the breakpoint. col. 10, lines 14-20. The Hervin patent, however, does not disclose or suggest various aspects of the claimed invention.

First, claims 1-15, 17-35, 37-40, 42, and 43 all recite or incorporate limitations directed to a minimum state of the executing service or saving a minimum state of the executing service. The Hervin patent, however, fails to disclose or suggest

these aspects of Applicants' claims. Specifically, the cited section of the patent specification merely explains that "write back WB stage updates the register file 24, condition codes, and other parts of the machine state with the results of the previously executed instruction." col. 6, lines 1-3. The Hervin patents makes no reference to a minimum state saving a minimum state. Rather, the Hervin patent merely discloses updating a register file based on a previously executed instruction. Moreover, the Hervin patent does not disclose or suggest a minimum state as recited in new claims 45-54.

Second, claims 1-12, 17-32, 37-39, and 42 all recite or incorporate limitations directed to altering a program counter of an executing service or a component used to alter a program counter. The Hervin patent, however, does not disclose or suggest these claim limitations. Rather, the Hervin patent refers to a breakpoint counter 106 or debug counter 106 stored in a debug configuration register. Thus, the counter disclosed in the Hervin patent is not of the executing service. Moreover, the Hervin patent fails to disclose or suggest a program counter that is altered. Rather, the cited patent merely discloses a counter that counts down from a predefined number X (as counters normally do), such that the exception does not occur until the Xth occurrence. See, e.g., col. 10, lines 17-55. Accordingly, the Hervin patent fails to disclose or suggest altering a program counter of an executing service or a component for altering a program counter.

Third, claims 1-12, 17, 37-39, and 42 all recite or incorporate limitations directed to restoring a program counter of the executing service. The Hervin patent fails to disclose or suggest these claim limitations since, as previously explained, the Hervin patent merely discloses a breakpoint counter of debug configuration register (compared to an executing service) that counts down from a predefined number X. col. 10, lines 18-20.

Fourth, claims 16, 17, 31, 36-38, 41, and 44 all recite or incorporate limitations directed to checking for a checksum error within the page of memory of the executing service. The Hervin patents fails to disclose or suggest these limitations. Rather, the cited sections of the specification relate to determining an optimum pipe for executing an instruction. There is no reference to a checksum error or checking for a checksum error in the cited section of the Hervin patent.

Fifth, claims 16, 17, 32, 36-38, 41, and 44 all recite or incorporate limitations directed to, if the executing service is set to reject the checksum error, saving the page of memory, inserting a breakpoint into the saved page of memory, or altering an instruction pointer, and processing the saved page of memory. The Hervin patent does not disclose or suggest these features since the cited patent does not disclose checking for a checksum error, much less performing the recited actions if the executing service rejects the checksum error.

Sixth, claims 13-15, 33-35, 40 and 43 all recite or incorporate limitations directed to setting a breakpoint at a last safe point and/or simulating instructions of the executing service from the last safe point to the breakpoint. The cited section of the Hervin patent, however, fails to disclose this feature of Applicants' claims. Rather, the cited section of the Hervin patent merely generally refers to setting a breakpoint, but makes no reference to setting a breakpoint at a last safe point. Accordingly, the cited patent does not disclose or suggest these limitations.

C. Conclusion

Based on the forgoing remarks, Applicants respectfully submit that the Hervin patent does not anticipate independent claims 1, 13, 16, 18, 33, 36, 39, 40, 42, and 43 of the subject application

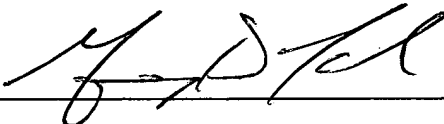
Application No. 09/539,197

since all of these claims include one or more of the previously discussed claim limitations. Further, Applicants' dependent claims 2-12, 14-15, 17, 19-32, 34-35, 37, 38, and 41 are also patentable over the cited reference, because they incorporate all of the limitations of independent claim 1. Additionally, new dependent claims 45-54 recite additional novel elements and limitations to their respective independent claims and are, therefore, also patentable over the cited reference.

Respectfully submitted,

CHRISTIE, PARKER & HALE, LLP

By



Gary D. Lueck
Reg. No. 50,791
626/795-9900

GDL/mac

MAC PAS474072.1-* -2/20/03 2:45 PM